NASA Technical Memorandum 80089

NASA-TM-80089 19790018639

TRENDS IN RELIABILITY MODELING TECHNOLOGY FOR FAULT TOLERANT SYSTEMS

SALVATORE J. BAVUSO

APRIL 1979

NASA
National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

TRENDS IN RELIABILITY MODELING TECHNOLOGY FOR FAULT TOLERANT SYSTEMS

Salvatore J. Bavuso
NASA Langley Research Center
Hampton, Virginia

SUMMARY

Recent developments in reliability modeling for fault tolerant avionic computing systems are presented. Emphasis is placed on the modeling of large systems where issues of state size and complexity, fault coverage, and practical computation are addressed.

A two-fold (analytical modeling) developmental effort is described based on the "structural modeling" and "fault coverage modeling" approaches. With regard to the structural modeling effort, two techniques under study are examined. One technique which was successfully applied to a 865 state pure death stationary Markov model is presented. The modeling technique is applied to a fault tolerant multiprocessor currently under development. Of particular interest is a short computer program which executes very quickly to produce reliability results of a large-state space model. Also, this model incorporates fault coverage states for processor, memory, and bus LRU's (Line Replaceable Unit).

A second structural reliability modeling scheme which is aimed at solving nonstationary Markov models is discussed. This technique which is under development will provide the tool required for studying the reliability of systems with nonconstant failure rates and includes intermittent/transient faults, electronic hardware which exhibits decreasing failure rates, and hydromechanical devices which typically have wearout failure mechanisms.

A general discussion of fault coverage and how it impacts system design is presented together with a historical account of the research which led to the current fault coverage developmental program. Several aspects of fault coverage including modeling and data measurement of intermittent/transient faults and latent faults are elucidated and illustrated. The CARE II (Computer-Aided Reliability Estimation) coverage is presented and shortcomings to be eliminated in the future CARE III are discussed.

The emergence of the so-called latent fault as a significant factor in reliability assessment is gaining increased attention from a modeling viewpoint; therefore, nuances of latent faults, models for such, and a method for latent fault measurement are depicted.

1. INTRODUCTION

The importance of achieving a faithful reliability assessment capability for avionic fault tolerant systems cannot be overstressed. Reliability issues involve virtually every aspect of design, packaging, and field operations, with regard to safety, maintainability, and invariably profits. Successful implementation of digital fault tolerant computers for critical flight functions in commercial aircraft cannot be realized without rigorous and credible analytical and simulative demonstrations of system reliability and fault tolerance. This conviction is fostered by the observation and supported by analysis that life testing to demonstrate the ultrareliability of these systems will be impractical, and because of the safety aspect, the full potential of such systems will not be realized until system reliability and fault tolerance are substantiated.

The task of producing a credible reliability assessment capability is indeed a formidable one. The root of the problem is embodied in the very essence that makes the digital computer such an attractive device for use in a host of applications, namely its adaptability to changing requirements, computational power, and ability to test itself.

Among the many factors to be considered in the design of fault tolerant systems are those which can have a direct impact on reliability. These factors must be accurately accounted for in a faithful reliability assessment. Figure 1 depicts some of the more important elements delineated into four categories: (1) Type and Manifestation, (2) Cause, (3) System Effect, and (4) Defense. Every digital avionic fault tolerant system must be designed to effectively cope with a myriad of hardware and software anomalies which are classified in categories 1 and 2. Categories 3 and 4 typify the effect of anomalies and some techniques for coping with them. Figure 2 portrays the combinations of categories 1 and 2. For example, a hardware anomaly could be a permanent random failure. On considering the number of devices in a digital system that are susceptible to failure in the ways depicted in figure 2 and combining software anomalies in a similar manner, one quickly begins to appreciate the designer's and the reliability analyst's tasks in accounting for these factors in reliability assessments. A rigorous discussion regarding some of these factors is given in McCluskey and Losq, 1978.

From a reliability assessment viewpoint, it was not until recently that analysts began to account for these factors (Roth et al., 1967) with the probabilistic concept of fault coverage. Since then, numerous reports have appeared on the effects of fault coverage accountability (Ultra-Systems, Inc., 1974; Bavuso, 1975; and Bjurman et al., 1976).

2. RELIABILITY MODELING APPROACH

Reliability modeling research at the NASA Langley Research Center has been strongly influenced by our fault tolerant computer architectural research program which commenced circa 1971 with the initiation of a study on the design of a Fault Tolerant Airborne Digital Computer (Wensley et al., 1973, and Ratner et al., 1973). This study identified two potentially viable computer architectures for aircraft flight control

applications. They are the SIFT (Software Implemented Fault Tolerance) and the FTMP (Fault Tolerant Multiprocessor) (Wensley et al., 1978, and Hopkins and Smith, 1975 and 1978). Both architectural concepts utilize multiple LSI (Large Scale Integration) processor and memory devices, resulting in a large number of SRU's (Smallest Reconfigurable Units). From a reliability modeling point of view, this scheme contributes heavily to the modeling complexity by increasing the number of possible operational hardware states. This state of affairs has focused our research in the direction of developing modeling techniques that are applicable to large-state models. For convenience, this modeling thrust will be referred to as the structural analytic approach. A parallel effort to the structural analytic approach was initiated by a study in 1973 which produced the Computer-Aided Reliability Estimation (CARE II) computer program. To date, the CARE II fault coverage model represents the most advanced generalized model published in the open literature. It was this study which launched the Langley fault coverage modeling approach.

Because it is anticipated that viable fly-by-wire digital fault tolerant systems for aircraft flight control will be required to meet unreliability requirements of (less than or equal to) 10^{-9} per flight and to be practical (less than or equal to) 10^{-9} at 10 hours, reliability models must be implemented in analytic form in lieu of simulation models; however, the use of very high speed emulators and/or parallel computers may at some future time diminish the analytic approach's dominance. This is not to say that simulative techniques are not presently applicable in reliability modeling. On the contrary, simulation plays a major role in determining vital reliability parameters associated with fault coverage modeling.

3. STATE-OF-THE-ART MODELING PROBLEMS

The state-of-the-art of structural analytic modeling of large systems is typified by the reliability analysis method employed in the ARCS (Airborne Advanced Reconfigurable Computer System) study (Bjurman et al., 1976). The solution technique is matrix oriented and is based on constructing a similarity relation such that the transition matrix is similar to a diagonal matrix containing the eigenvalues along the diagonal. For pure-death Markov processes with distinct eigenvalues, this solution method is extremely fast in a general purpose digital computer and, thus, very attractive for use in large-state space models. With some minimal care in assigning failure rate data so that, for all practical purposes, the system eigenvalues are mathematically distinct, this solution scheme is applicable to a large class of computer architectures of practical interest. Such a system is the FTMP which was analyzed at Langley using the described method. An abbreviated state transition diagram for the FTMP appears in figure 3 where a system state is defined as the 6-tuple vector, (a,b,d,c,e,f), where

a = number of working processor modules
b = number of processor modules in a recovery state
c = number of working memory modules
d = number of memory modules in a recovery state
e = number of working bus modules
f = number of bus modules in recovery

and the SRU's are the processor, memory, and bus modules. Initially the system is in state (10, 0, 10, 0, 5, 0) and the final state is (5, 0, 2, 0, 2, 0). Further loss of hardware is considered system failure since crucial flight functions cannot be effected. Elements, b, d, and f describe states involving recovery. In addition to system loss resulting from hardware depletion, system failure occurs (in this model) when a second fault occurs within a recovery interval. This condition was imposed because the FTMP's primary fault detection and isolation mechanisms are based on a functional level software majority voting scheme. In actuality, the FTMP can recover from many double failures; however, the double failure constraint was necessary to reduce the state size of the reliability model; fortunately, it also produces a conservative reliability estimate. Several other necessary conservative assumptions were required to bring the state size down to a manageable level; in this case, a 865-state model resulted. Although 865 states for a reliability model is considered very large by industry standards, this analysis presented no problem for our Control Data Corporation CYBER 175 computer. In fact, a mission time of 10 hours required only 74 CPU seconds.

Aside from the surprising low CPU time of such a complex analysis, another unexpected outcome resulted and is shown in figure 4. The probability of system failure in 10 hours is plotted against processor failure rate per hour for the 865-state model with 10 processors, 10 memories, and 5 buses; and for a 673-state model with 10 processors, 8 memories, and 5 buses. The data show that the addition of 2 memory modules increased the system probability of failure. This trend also applies if in lieu of "processor" appearing in figure 4, "memory" or "bus" is plotted. One explanation for this unexpected data is the sensitivity of the reliability model to the occurrence of a second fault during recovery. Beyond a particular hardware complement, increasing hardware redundancy diminishes system reliability because of the increased likelihood of additional faults. If the constraint that a second fault occurring in a recovery interval fails the system were relaxed, the results will change in favor of increasing redundancy. The penalty for increased realism is a considerable increase in the model state size. To date, a practical upper bound on the state size for the matrix solution technique previously discussed has not been explored. On the pessimistic side, it is sobering to realize that the 865-state model was reduced from approximately 10 million states through the imposition of certain conservative constraints on the model.

The state-of-the-art of reliability modeling of large systems has progressed one step beyond that already described to include transient faults. This amounts to adding the transient failure rate (transition rate) to hardware failure rates to account for persistent transient faults that behave like permanent faults (Bjurman et al., 1976, and Ng, 1976). The reliability contribution due to the time the machine spends in the recovery state because of a transient is not accurately modeled: As most analyses assume constant transient transition rates, one can ignore the recovery state and combine the transient transition rate with the permanent fault transition rate.

This scenario of the state-of-the-art of reliability modeling for fault tolerant systems surely must convey the notion of modeling inaccuracies, not to mention the conspicuous absence of any discussion of software anomalies and other anomalies portrayed in figure 2. Even though the reliability analyst makes

every attempt to be conservative when he cannot be accurate, more often than not he is forced into a compromising position that raises doubt and diminishes confidence in the analysis.

4. A NOVEL APPROACH FOR RELIABILITY ASSESSMENT

In the aforementioned analysis of the FTMP, irrespective of software considerations, the major suspects which challenge both the accuracy and the conservatism of the analysis are the transient and fault recovery treatment. In both cases, it was assumed that the state transition rates are constant and values for these were determined by educated guesses. Also, it was assumed that the latency time is zero (Shedletsky and McCluskey, 1976). Trends in reliability modeling technology for fault tolerant systems are being driven by the need for analytic techniques capable of modeling fault tolerant systems with state sizes on the order of 1000, to include sensors, actuators, and their computer interfaces. There is mounting evidence that certain electronic devices exhibit nonconstant hazard rates (Timming, 1975, and Shooman, 1974); and mechanical and hydraulic devices commonly exhibit wearout, i.e., increasing hazard rates with time. These observations coupled with the need to accurately account for fault latency, intermittent/transient faults, and software failures present a strong case for an analytic technique capable of modeling nonconstant hazard rates.

The development of such a technique is currently under study and will result in the development of a General Computer-Aided Reliability Estimation (CARE III) computer program. The desire to reduce the large state sizes for Markov processes vis-a-vis CARSRA (Computer-Aided Redundant System Reliability Analysis, Bjurman et al., 1976) and the need to treat nonconstant hazard rates directed the study toward a generalized Markov process concept, namely the processes in which the Chapman-Kolmogorov equation holds:

$$P_{\ell i}(t,\tau) = \sum_{\nu} P_{\nu i}(s,\tau) P_{\ell \nu}(t,s)$$

for all $\tau < s < t$, where $P_{\ell i}(t,\tau)$ is the probability that the system is in state ℓ at time t given that it was in state i at time τ (Feller, 1957). By judicially defining system states to satisfy the Chapman-Kolmogorov equation, the forward Kolmogorov equation can be satisfied under some very general conditions:

$$\frac{\partial P_{\ell i}(t,\tau)}{\partial t} = -P_{\ell i}(t,\tau)\,\lambda_{\ell i}(t,\tau) + \sum_{j \neq \ell} P_{ji}(t,\tau)\, C_{j\ell i}(t,\tau)\lambda_{j\ell i}(t,\tau)$$

If the notation indicating the condition that the system be in state i at time τ be suppressed, the following recursive equation results:

$$P_{\ell}(t) = e^{-\int_0^t \lambda_{\ell}(\tau)d\tau} \int_0^t \frac{\sum_j P_j(\tau) c_{j\ell}(\tau)\lambda_{j\ell}(\tau)}{e^{-\int_0^\tau \lambda_{\ell}(\eta)d\eta}}\, d\tau$$

where

$P_{\ell}(t)$ = probability of being in state ℓ at time t

$\lambda_{j\ell}(t)$ = transfer rate from state j to state ℓ

$\lambda_{\ell}(t) = \sum_j \lambda_{\ell j}(t)$

$c_{j\ell}(t)$ = coverage associated with a failure which, if coverage were perfect, would cause a transfer from state j to state ℓ

The system reliability is given by

$$R(t) = \sum_{\ell \in L} P_{\ell}(t)$$

for the set L of allowable states.

From a computational point of view, a more accurate form is obtained by letting

$$Q_{\ell}(t) = P^{*}_{\ell}(t) - P_{\ell}(t)$$

where $P^{*}_{\ell}(t) = P_{\ell}(t)$ given perfect coverage. The system unreliability $Q(t)$ is given by

$$Q(t) = 1 - R(t) = \sum_{\ell \in L} Q_\ell(t) + \sum_{\ell \in \bar{L}} P^*_\ell(t)$$

with $\bar{L} \cup L$ being the set of all possible states. And $\bar{c}_{j\ell}(t) = 1 - c_{j\ell}(t)$ so that

$$Q_\ell(t) = e^{-\int_0^t \lambda_\ell(\tau)d\tau} \int_0^t \frac{\left[\sum_j Q_j(\tau) + P_j(\tau)\bar{c}_{j\ell}(\tau)\right]\lambda_{j\ell}(\tau)}{e^{-\int_0^\tau \lambda_\ell(\eta)d\eta}} d\tau$$

The virtues of this scheme are that the hazard rate $\lambda_{j\ell}(t)$ and coverage $c_{j\ell}(t)$ are time dependent; also the contribution to system unreliability due to perfect and imperfect coverage is decoupled. The need for the $\lambda_{j\ell}(t)$ was previously discussed, but the importance of $c_{j\ell}(t)$ was not presented.

In avionic systems which utilize dynamic resource allocation schemes such as is possible with the FTMP and SIFT systems, the proportion of hardware and software resources is dependent on the aircraft flight phase and/or flight envelope. Flight critical phases require greater hardware redundancy and fault monitoring. The latter factor appears in reliability models as time-varying coverage $c_{j\ell}(t)$. A more subtle need for $c_{j\ell}(t)$ is to account for fault latency. The probability of system failure due to insufficient coverage is a function of the number of existing failures embedded in the system. That is, the probability of a second SRU (processor, bus, memory) failure occurring during the τ second recovery time is a function of the number of SRU's functioning at that time.

Preliminary studies of the Kolmogorov technique are encouraging from an accuracy viewpoint and computer run time. Figures 5 and 6 compare FTMP reliability data generated with the Kolmogorov technique against data generated with other conventional techniques. To make a meaningful comparison, $c_{j\ell}(t)$ and $\lambda_{j\ell}(t)$ were constrained as constants in the Kolmogorov technique. It is suspected that the discrepancies depicted in figure 6 are attributed to simplifying assumptions required to keep the conventional analysis technique tractable.

Current work on CARE III is directed toward developing a coverage model compatible with the Kolmogorov technique and is based to a large extent on the CARE II coverage model (Raytheon Company, 1974 and 1976). Improvements to be sought are modification for coverage time dependency ($c_{j\ell}(t)$) to model latent faults and of greater difficulty, to reduce the burden placed on the user in defining input data for the modified CARE II coverage model. A third improvement is to include a more sophisticated intermittent/transient fault coverage model and if possible a software failure model.

The CARE II coverage model is a powerful basis upon which to build the Kolmogorov coverage model (KCM). In its completed form, the KCM will determine coefficients for the Kolmogorov reliability model (KREL-M). Coverage is conceived as consisting of three fundamental processes, system fault detection, fault isolation to the SRU, and recovery, which may require hardware replacement and/or software correction. Failure to properly effect one of these processes constitutes a coverage failure which is usually modeled as a system failure. A faithful coverage model must provide the mechanisms by which the reliability analyst can relate the coverage coefficients to the system factors that affect coverage. These factors include the fault classes (permanent/intermittent hardware/software faults), the system fault detection mechanisms (software/hardware voting, software self-monitoring, BITE (Built In Test Equipment), etc.), SRU fault isolation mechanisms (similar to detection), and recovery procedures (hardware replacement, instruction retry, etc.). Detectors are modeled as competitors in the detection process. Every detector has some chance of discovering a fault; however, most detectors usually are specialized for a particular class of faults. In CARE II, this modeling process is under user control. It is assumed in the coverage model that the detector which discovers a fault is most capable of defining fault isolation and recovery strategies. These strategies are user defined.

The CARE II coverage model takes the following form:

$$c_x(i,j) = P_i P_{sx}{}^{j-1} P_i' \int_0^\infty \int_0^\infty g_i(\tau) h_i(\tau' - j\tau_{sx}) r_i(\tau,\tau') d\tau\, d\tau'$$

where

$c_x(i,j)$ = conditional probability system can recover from a fault in stage x given the fault belongs to fault class j and is detected by detector i *

τ = detection time

τ' = isolation time

P_{sx} = defective spare detection

τ_{sx} = spare unit test time

*A stage is defined as a set of identical devices.

P_i = noncompetitive detection probability

P_i' = isolation probability associated with P_i

h_i = isolation rate

r_i = recovery probability

g_i = competitive detection rate

Of all of these parameters, $g_i(\tau)$ is the most difficult to obtain because it is a function of detector i and the entire ensemble of detectors and their interrelationships.

5. ACQUISITION OF COVERAGE DATA

Assuming success in modifying the CARE II coverage model for the KREL-M, some difficulty in using this capability still remains. Eventually the analyst must obtain coverage data peculiar to the system of interest. Three types of data are urgently required: intermittent hazard rate data including duration densities, fault detection densities for various classes of faults and detectors, and software hazard rate data. There is some encouraging news on the first two; a discussion on the third is beyond the scope of this paper and will not be addressed further.

A source of intermittent arrival data has been identified and work has recently commenced to generate a data base of intermittent field hardware failure data in digital electronics*. The long-term aim of this endeavor is to produce intermittent hazard rate data for a variety of digital devices using different parts technology but applicable to avionics.

Beyond the pressing issues surrounding software reliability, validity and/or validation, characterization of the latent fault ranks in equal importance to the eventual success of utilizing digital systems for flight critical functions. Because of the near infinite number of possible machine states that a digital computer can obtain as a result of failures, it is impossible to exhaustively test such a device to determine its health. Therefore the presence of undetected faults is always a possibility, and for systems designed to obtain system probabilities of failure of less than 10^{-9} in 10 hours of flight, even small probabilities of latent faults occurring can have a large effect on system reliability. It is certainly with these thoughts in mind that designers incorporate redundancy; however, the cost of constructing machines which tolerate more than three coexisting manifested faults becomes prohibitive. An acceptable solution is to constantly search for faults and eliminate their effects so that the machine is never presented with two coexisting manifested faults, i.e., only one at a time. To insure that this goal is satisfied, the designer must have a priori knowledge of fault occurrence and manifestion rates so that adequate fault detection and recovery mechanisms can be incorporated.

There are a number of detection schemes; the most obvious is comparison/voting and can be implemented in at least one of two ways: by executing a special software test and comparing expected results with computed results (self-monitoring) or two or more uniprocessors can compare functional level outputs during normal computation where both processors are executing the same code. The time between fault occurrence and its detection is the latency time. If this time is short compared with the failure rate of SRU's, then the machine will essentially see single failures and have sufficient time to cope with them. Long latency times are conducive to system failure.

In an attempt to determine methods of acquiring latency data, a study entitled, "Modeling of a Latent Fault Detector in a Digital System" was conducted (Nagel, 1978). A very simple computer (VSC) modeled at the gate level was designed and simulated to execute on a CDC CYBER 175 host computer. Six simple programs were written using the VSC that consisted primarily of the following instructions:

Fetch and store
Add and subtract
Shift right and shift left
AND and OR
Indirect addressing
Overflow indicator
Branch
Copy to and from temporary storage

While the VSC executed each of the six programs, single faults were induced random uniformly over the gate list. Input, output, stuck-at-one, and stuck-at-zero faults were equally likely occurrences. Initially the number of runs manifesting faulty output was recorded and produced the following results:

PROGRAM	SAMPLE SIZE	DETECTIONS	ESTIMATED DETECTION PROBABILITY	ESTIMATED STANDARD DEVIATION
Fibonacci (FIB)	211	98	0.464	0.034
Fetch and Store (F&S)	118	42	.356	.044
Add and Subtract (A&S)	208	117	.563	.034
Search and Compute	118	64	.542	.046
Linear Convergence	133	78	.586	.043
Quadratic	97	55	.577	.050

*NASA Contract Number, NAS1-15574 with Sperry Univac.

Extensive data analysis was performed to explain the observed differences in terms of the number of executed instructions, the number of different instructions used in computation, the degree of branching, the fault mode (stuck-at-one or zero, input or output), and number size. The results of the statistical analyses indicate that latency time, or equivalently, detection capability, depends primarily on the instruction subset used during computation and the frequency of its use. Moreover, little direct dependence was observed for such factors as fault mode, number size, degree of branching, and program length. An exponential model was proposed and applied to the data from three programs (Add and Subtract, Fibonacci, and Fetch and Store).

The exponential model is based on the density function of $y = \min(t, T)$, where t is the detection time measured in repetitions and T is the truncation time of test, and is given by:

$$f(y) = \begin{cases} P_o \lambda e^{-\lambda y} & y < T \\ P_o e^{-\lambda T} + Q_o & y = T \\ 0 & \text{Elsewhere} \end{cases} \qquad (Q_o = 1 - P_o)$$

where

P_o = the detection probability

Q_o = the probability of nondetection for all time

$P_o e^{-\lambda T}$ = the probability of nondetection due to insufficient test time

Values for P_o and λ were obtained using maximum likelihood estimators, enabling the following data to be generated:

Program	P_o	λ	λ/P_o
A&S	0.568	0.577	1.02
FIB	.474	.491	1.04
F&S	.371	.398	1.07

A pictorial representation of this model is shown in figure 7 superimposed on the raw data in histogram form.

If after careful testing, this method of measuring and modeling fault latency proves to be acceptable, an important set of coverage parameters will become available for reliability modeling. As an aside, this scheme also provides a method for synthesizing test programs both for pre-flight and in-flight monitoring.

6. CONCLUDING REMARKS

Testing digital systems which perform flight critical functions is not a feasible method for estimating system reliability. Analytic modeling of system reliability in conjunction with simulative techniques for coverage measurement appears to be the only alternative on the horizon. Accurate reliability estimates which account for such factors as latent faults, intermittent/transient faults, and software errors require sophisticated techniques which are currently being developed and will result in the KREL-M reliability assessment capability embodied in the CARE III computer program. The effects and significance of these factors on the reliability of fault tolerant digital systems are yet to be determined, and the potential of increased complexity brought about by the inclusion of these factors in an assessment capability such as KREL-M is a major concern. It is anticipated that after extensive trade-off analyses, KREL-M will be simplified and take on more of the characteristics of a production tool in lieu of its initial experimental character.

In a parallel effort, methods for acquiring indispensable coverage data required by KREL-M are now becoming available.

REFERENCES

Bavuso, S. J., 1975, Impact of Coverage on the Reliability of a Fault Tolerant Computer, NASA TN D-7938.

Bjurman, B. E., Jenkins, G. M., Masreliez, C. J., McClellan, K. L., and Templeman, J. E., 1976, Airborne Advanced Reconfigurable Computer System (ARCS), The Boeing Commercial Airplane Company, NASA CR-145024.

Feller, W., 1957, An Introduction to Probability Theory and Its Applications, John Wiley & Sons, Inc.

Hopkins, A. L., and Smith, T. B., 1975, The Architectural Elements of a Symmetric Fault-Tolerant Multiprocessor, IEEE Trans. on Computers, vol. C-24, no. 5.

Hopkins, A. L., and Smith, T. B., 1978, A Fault Tolerant Multiprocessor Architecture for Aircraft, Vol. I, The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts, NASA CR-3010.

McCluskey, E. J., and Losq, J., 1978, Critical Fault Patterns Determination in Fault-Tolerant Computer Systems, Stanford University, NASA CR-145352.

Nagel, P. M., 1978, Modeling of a Latent Fault Detector in a Digital System, NASA CR-145371.

Ng, Y., 1976, Reliability Modeling and Analysis for Fault-Tolerant Computers, PhD Dissertation, UCLA - EWG-7698.

Ratner, R. S., et al., 1973, Design of a Fault Tolerant Airborne Digital Computer, Volume II - Computational Requirements and Technology, Stanford Research Institute, Menlo Park, California, NASA CR-132253.

Raytheon Company, Sudbury, Massachusetts, 1974, Reliability Model Derivation of a Fault-Tolerant, Dual, Spare-Switching Digital Computer System, NASA CR-132441.

Raytheon Company, Sudbury, Massachusetts, 1976, An Engineering Treatise on the CARE II Dual Mode and Coverage Models, NASA CR-144993.

Roth, J. P., Bouricius, W. G., Carter, W. C., and Schneider, P. R., 1967, Phase II of an Architectural Study for a Self-Repairing Computer, SAMSO TR-67-106, United States Air Force. (Available from DDC as AD 825460.)

Shedletsky, J. J., and McCluskey, E. J., 1976, The Error Latency of a Fault in a Sequential Digital Circuit, IEEE Trans. on Computers, vol. C-25, no. 6.

Shooman, M. L., 1974, Hazard Function Monitoring of Airline Components, Proceedings of Annual Reliability and Maintainability Symposium.

Timming, A. R., 1975, A Study of Total Space Life Performance of GSFC Spacecraft, NASA TN D-8017.

Ultra-Systems, Inc., Newport Beach, California, 1974, Reconfigurable Computer Systems Study, NASA CR-132537.

Wensley, J. H., et al., 1973, Design of a Fault Tolerant Airborne Digital Computer, Volume I - Architecture, Stanford Research Institute, Menlo Park, California, NASA CR-132252.

Wensley, J. H., et al., 1978, Design Study of Software - Implemented Fault Tolerance (SIFT) Computer, SRI International, Menlo Park, California, NASA CR-3011.

CATEGORY		
1. TYPE & MANIFESTATION:	- HARDWARE ANOMALY ● PERMANENT ● TRANSIENT ● INTERMITTENT	- SOFTWARE ANOMALY ● PERMANENT ● TRANSIENT ● INTERMITTENT
2. CAUSE:	- DESIGN ERROR - FABRICATION ERROR - RANDOM FAILURE - EXTERNALLY INDUCED ● SIGNAL ERROR ● POWER FAILURE ● PHYSICAL FAILURE ● EMI	- DESIGN ERROR - CODING ERROR - EXTERNALLY INDUCED ● DATA PATTERN ERROR ● PROCEDURE ERROR
3. SYSTEM EFFECT:	- COMPUTER SYSTEM CONTROL LOSS - APPLICATION COMPUTATION ERROR - NONE	- COMPUTER SYSTEM CONTROL LOSS - APPLICATION COMPUTATION ERROR - NONE
4. DEFENSE	- HARDWARE REDUNDANCY ● SPATIAL - ALTERNATE HARDWARE ● TEMPORAL - RETRY (TRANSIENT)	- SOFTWARE REDUNDANCY ● SPATIAL - ALTERNATE CODE ● TEMPORAL - RETRY

Figure 1. Factors affecting coverage.

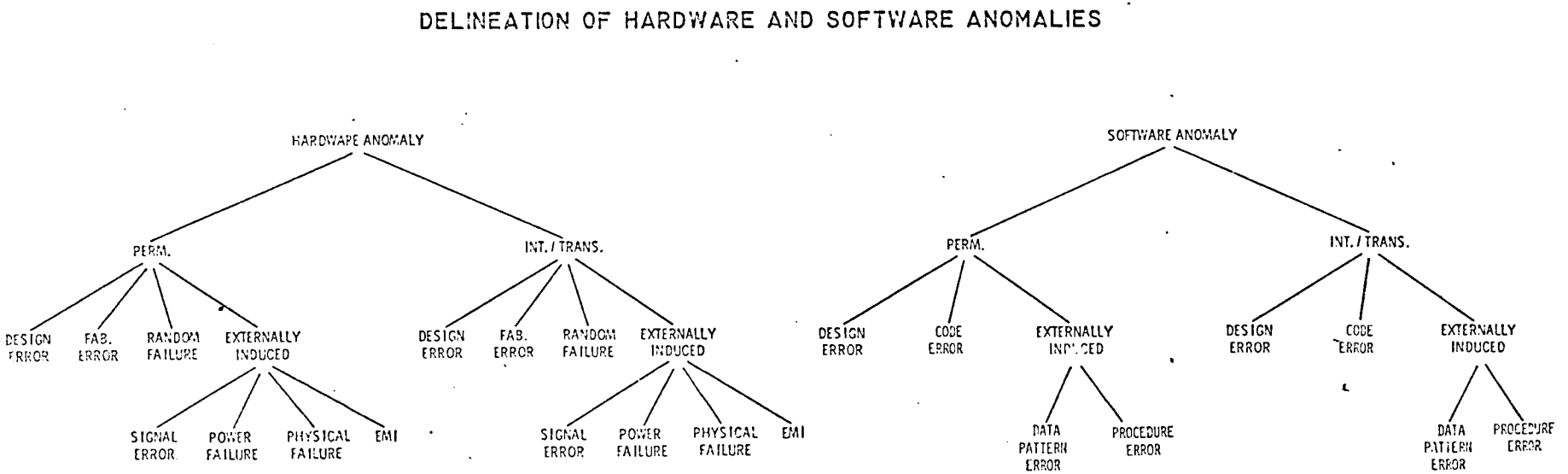

Figure 2. Delineation of hardware and software anomalies.

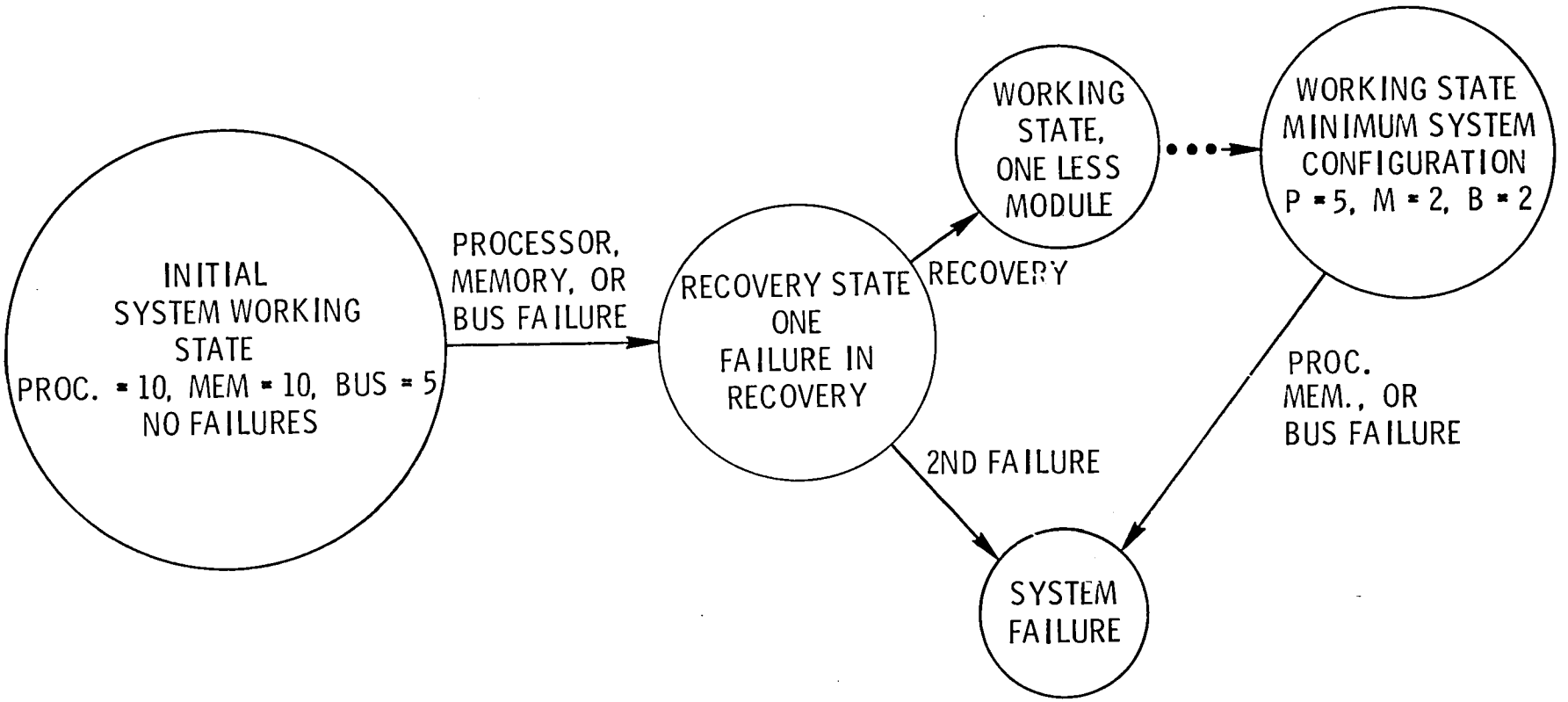

Figure 3. FTMP state transition diagram.

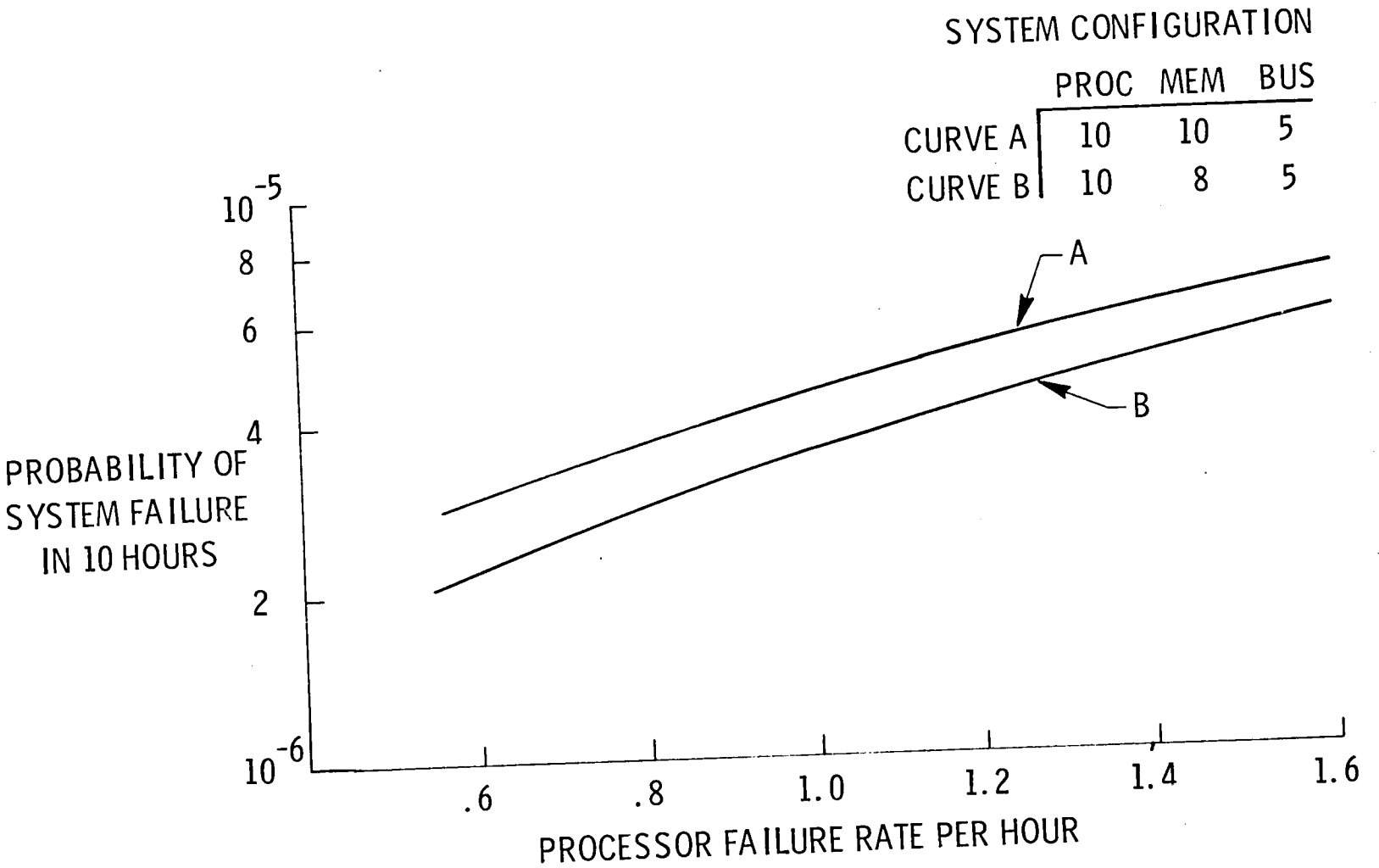

Figure 4. Probability of system failure versus processor failure rate for the FTMP.

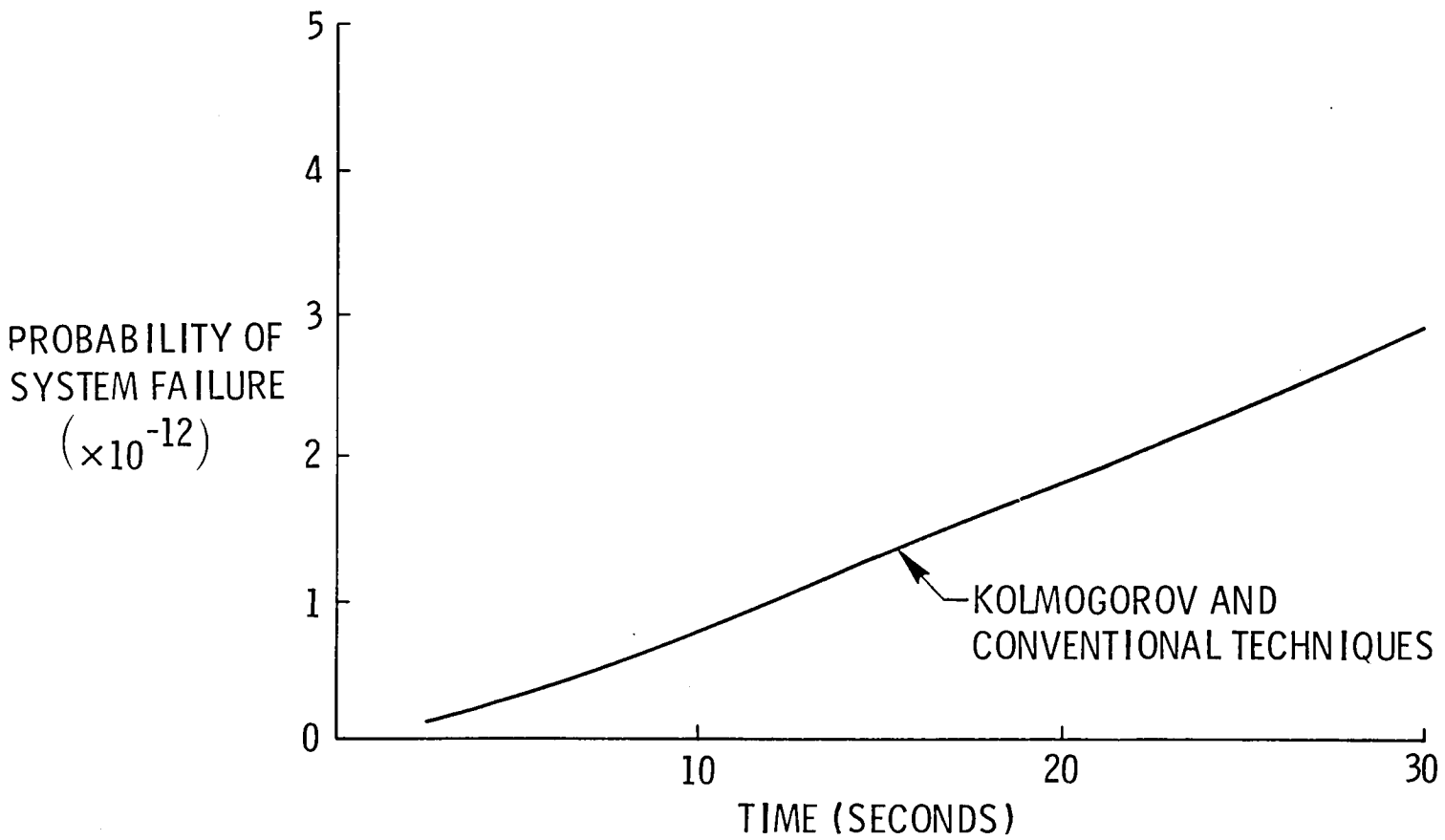

Figure 5. Probability of system failure versus time for FTMP.

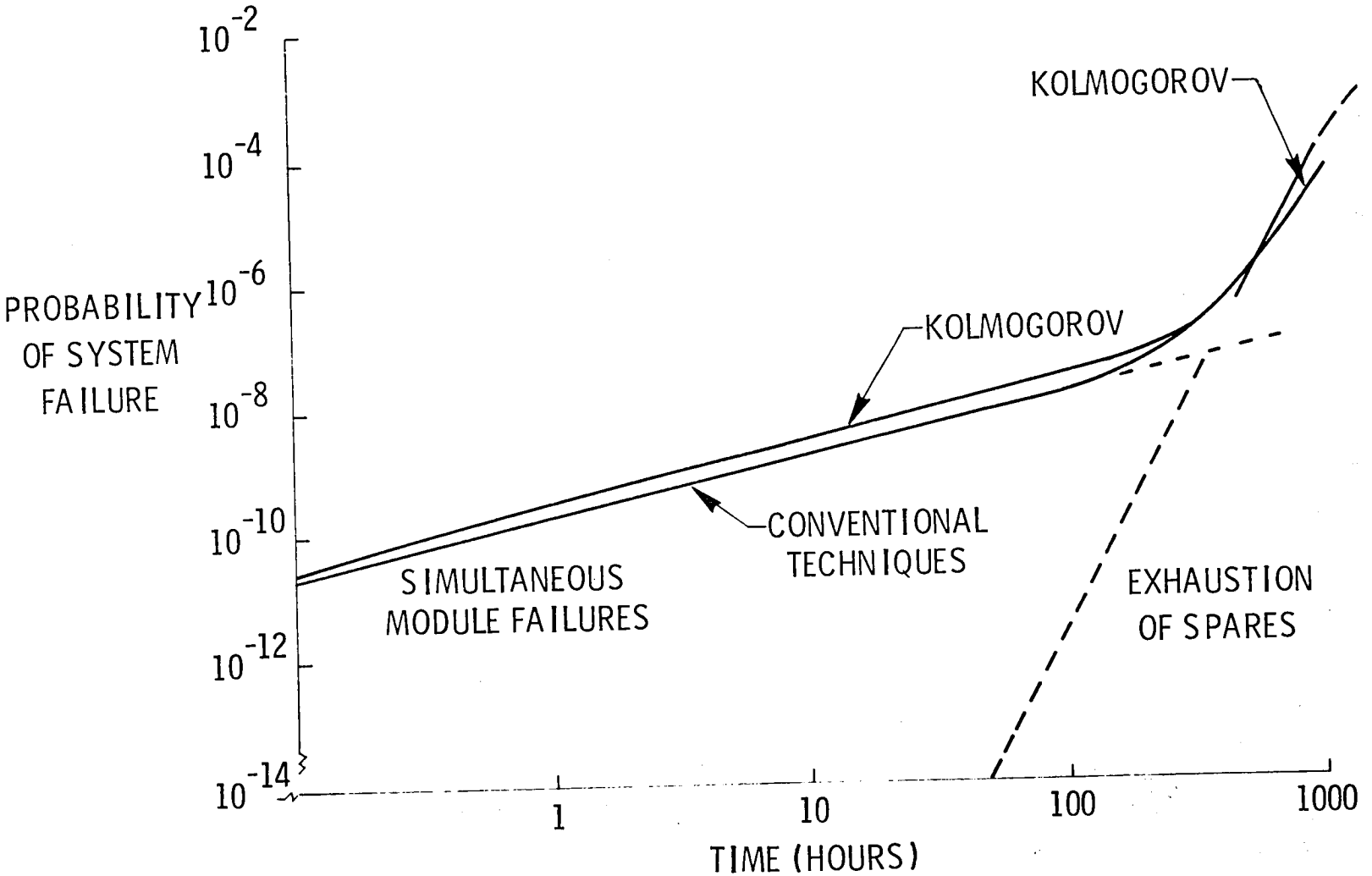

Figure 6. Probability of system failure versus operating time for the FTMP.

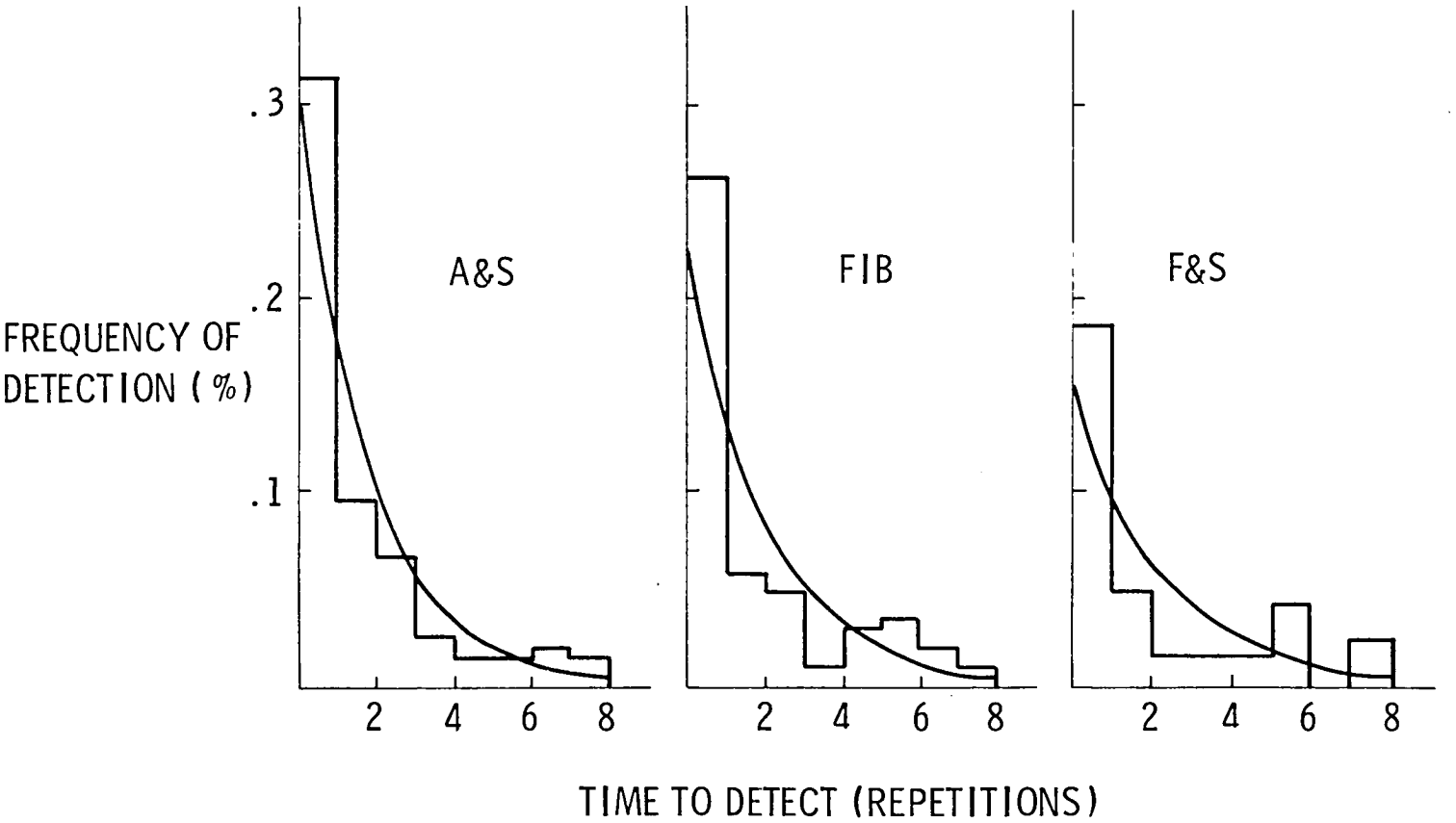

Figure 7. Exponential model of fault latency (detection).

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-80089		

4. Title and Subtitle	5. Report Date
TRENDS IN RELIABILITY MODELING TECHNOLOGY FOR FAULT TOLERANT SYSTEMS	April 1979
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
SALVATORE J. BAVUSO	
9. Performing Organization Name and Address	10. Work Unit No.
Langley Research Center Hampton, Virginia 23665	505-07-33-01
	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Recent developments in reliability modeling for fault tolerant avionic computing systems are presented. Emphasis is placed on the modeling of large systems where issues of state size and complexity, fault coverage, and practical computation are addressed.

State-of-the-art reliability modeling techniques for fault tolerant computing systems are discussed in conjunction with a novel technique currently under development. The latter capability will provide the tool required for studying the reliability of systems with nonconstant failure rates including intermittent/transient faults and latent fault models.

A study aimed at measuring fault latency is discussed which may provide a method of obtaining vital latent fault data.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
RELIABILITY, FAULT COVERAGE, AND EMULATION	Unclassified - Unlimited Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
UNCLASSIFIED	UNCLASSIFIED	14	$4.00

* For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document